Parameter Control in Differential Evolution for Constrained Optimization

Efrén Mezura-Montes, *Member, IEEE* and Ana Gabriela Palomeque-Ortiz

***Abstract*— In this paper we present the addition of parameter control in a Differential Evolution algorithm for constrained optimization. Three parameters are self-adapted by encoding them within each individual and a fourth parameter is controlled by a deterministic approach. A set of experiments are performed in order (1) to determine the performance of the modified algorithm with respect to its original version, (2) to analyze the behavior of the self-adaptive parameter values and (3) to compare it with respect to state-of-the-art approaches. Based on the obtained results, some findings regarding the values for the DE parameters as well as for the parameters related with the constraint-handling mechanism are discussed.**

I. Introduction

The optimization process consists on finding the best solution for a given problem under certain conditions. Nowadays, the use of alternative approaches to solve complex optimization problems is very common [1]. Evolutionary Algorithms (EAs) have received many interest from researchers and practitioners due to their competitive results when solving this kind of problems [2].

This paper will focus on the numerical optimization problem with constraints. Without loss of generality, it can be defined as to: Find $\vec{x}$ which minimizes $f(\vec{x})$ subject to $g_i(\vec{x}) \leq 0, \;\; i = 1, \ldots, m$ $h_j(\vec{x}) = 0, \;\; j = 1, \ldots, p$ where $\vec{x} \in \mathbb{R}^n$ is the vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ and each $x_i, \;\; i = 1, ..., n$ is bounded by lower and upper limits $L_i \leq x_i \leq U_i$ which define the search space $\mathcal{S}$. $\mathcal{F}$ comprises the set of all solutions which satisfy the constraints of the problems and it is called the feasible region; m is the number of inequality constraints and p is the number of equality constraints (constraints could be linear or nonlinear). Equality constraints are transformed into inequalities constraints as follows: $|h_j(\vec{x})| - \varepsilon \leq 0$, where ε is the tolerance allowed (a very small value).

EAs, in their original versions, lack a mechanism to incorporate feasibility information in the fitness value of individuals. A considerable amount of research has been dedicated to design techniques to handle the constraints of a given optimization problem [3], [4].

Constraint-handling techniques can be divided in two groups: (1) Those based on penalty functions i.e. a combination of the objective function value and the sum of constraint violation and (2) those based on the separated use of the objective function value and the sum of constraint violation in the fitness value of a solution.

Differential Evolution (DE) [5], [6], shares similarities with previous EAs. For example, DE works with a population of solutions, called vectors, it uses recombination and mutation operators to generate new vectors and, finally, it has a replacement process to discard the less fit vectors. DE uses real encoding to represent solutions. Some of the differences with respect to other EAs are the following: DE does not use a pre-defined probability distribution for its mutation operator, such as Gaussian distribution. Instead, DE uses the current distribution of vectors in the population to define the behavior of the mutation operator.

As other EAs, DE requires the careful fine-tuning of their parameter values. In [7] a classification of parameter setting techniques is presented: (1) Parameter tuning and (2) parameter control. Parameter tuning consists on defining good values for the parameters before the run of an algorithm and, then, running it with these values. On the other hand, three different ways to control parameter values are considered: (a) deterministic parameter control aims to modify the parameter values by a deterministic rule e.g. a fixed schedule, (b) adaptive parameter control aims to modify the parameter values based on some feedback from the search behavior e.g. diversity measure to update the mutation rate and (c) self-adaptive parameter control encodes the parameter values into the chromosomes of solutions and they are subject to variation operators. Most of the work related to parameter setting is focused on variation operators (e.g. mutation) and population size [7].

DE for constrained optimization has been studied previously. Lampinen used DE to tackle constrained problems [8] by using Pareto dominance in the constraints space. Mezura et al. [9] proposed to add Deb's feasibility rules [10] into DE to deal with constraints. Kukkonen & Lampinen [11] improved its DE-based approach to solve constrained multiobjective optimization problems. Zielinsky & Laur also used Deb's rules [10] in DE to solve some constrained optimization problems [12]. There are also hybrid approaches such as the DE with gradient-based mutation by Takahama & Sakai [13], [14] and PSO-DE (PESO+) by Muñoz-Zavala et al. [15]. Finally, other authors have proposed novel DE variants for constrained optimization [16] or multi-population-based DE approaches [17].

On the other hand, there are some studies regarding parameter control in DE for constrained optimization. Brest et al. [18] have proposed an adaptive parameter control for two DE parameters related to the crossover and mutation

Efrén Mezura-Montes and Ana Gabriela Palomeque-Ortiz are with the National Laboratory on Advanced Informatics. Rébsamen 80, Centro, Xalapa, Veracruz, 91000, MEXIC0. Email (emezura@lania.mx, apalomeque@lania.edu.mx)

operators. Huang et al. [19] used an adaptive mechanism to select among a set of DE variants to be used for the generation of new vectors based on a success measure. Moreover, some DE parameters to control the variation operators were also adapted. Finally, Liu & Lampinen [20] proposed to adapt DE parameters by means of Fuzzy Logic.

In this work, we present different parameter control techniques applied to a competitive DE-based algorithm to solve constrained optimization problems. Unlike previous proposals found in the specialized literature, our approach considers: (1) the adaptation of parameters related with the constraint-handling technique (no penalty functions are used in this work) and (2) an analysis of the behavior of the parameter values during the process.

The paper is organized as follows: In Section II DE is briefly introduced. Section III presents Diversity Differential Evolution, the algorithm used in our approach and it also includes the details of the parameter control techniques proposed. After that, in Section IV the experimental design, the results obtained and the corresponding discussion are included. The paper ends with Section V, where the conclusions and future work are shown.

II. Differential Evolution

DE is a simple, but powerful algorithm that simulates natural evolution combined with a mechanism to generate multiple search directions based on the distribution of solutions in the current population. Each vector i in the population at generation G, $\vec{x}_{i,G}$, called *target vector* will generate one offspring, called *trial vector* ($\vec{u}_{i,G}$). The trial vector is generated with the following process: A search direction is defined by calculating the difference between a pair of vectors, called *differential vectors*, both of them chosen at random from the population. This difference vector is also scaled by using a user-defined parameter called $F \geq 0$ [5]. This scaled difference vector is then added to a third vector, called *base vector*. As a result, a new vector is obtained, known as the *mutation vector*. After that, this mutation vector is recombined with the target vector (also called parent vector) to generate a *trial vector* (child vector) by using discrete recombination (usually binomial crossover) controlled by a crossover parameter $0 \leq CR \leq 1$ whose value determines how similar this trial vector will be with respect to the mutation vector. There are several DE variants [5] and some of them require the definition of extra parameters e.g. K for some types of crossover operators. However, the most known and used is DE/rand/1/bin, where the base vector is chosen at random, there is only a pair of differential vectors and a binomial crossover is used. The detailed pseudocode of this variant is presented in Figure 1.

III. Our approach

Research in parameter control for constrained optimization is scarce compared to unconstrained optimization. Furthermore, the research efforts usually do not consider, with the exception of penalty-function-based approaches (as in [21]), the parameters added with the constraint-handling mechanism. The goal of this work is to propose parameter control mechanisms in a competitive EA for constrained optimization by considering parameters of the constraint-handling technique. Furthermore, an analysis of the behavior of the parameters is taken into account.

Begin
G=0
Create a random initial population $\vec{x}_{i,G}$ $\forall i$, $i = 1, \ldots, NP$
Evaluate $f(\vec{x}_{i,G})$ $\forall i$, $i = 1, \ldots, NP$
For G=1 to MAX_GEN **Do**
For i=1 to NP **Do**
Select randomly $r_1 \neq r_2 \neq r_3 \neq i$
j_{rand} = randint$(1, D)$
For j=1 to n **Do**
If ($rand_j[0,1) < CR$ **or** $j = j_{rand}$) **Then**
$u_{i,j,G+1} = x_{r_3,j,G} + F(x_{r_1,j,G} - x_{r_2,j,G})$
Else
$u_{i,j,G+1} = x_{i,j,G}$
End If
End For
If ($f(\vec{u}_{i,G+1}) \leq f(\vec{x}_{i,G})$) **Then**
$\vec{x}_{i,G+1} = \vec{u}_{i,G+1}$
Else
$\vec{x}_{i,G+1} = \vec{x}_{i,G}$
End If
End For
$G = G + 1$
End For
End

Fig. 1. "DE/rand/1/bin" pseudocode. rand$[0,1)$ is a function that returns a real number between 0 and 1. randint(min,max) is a function that returns an integer number between min and max. NP, MAX_GEN, CR and F are user-defined parameters. n is the dimensionality of the problem.

We use a competitive approach in constrained optimization, known as Diversity Differential Evolution (DDE) [22], where the constraint-handling mechanism used adds some parameters, which are now considered for parameter control in the current research. The behavior of each parameter during the evolutionary process is analyzed in order to get more knowledge about the way DE is solving the constrained problem.

DDE, detailed in Figure 2, modifies traditional DE as follows [22]:

1) The probability of a target vector to generate a better trial vector is increased by allowing it to generate NO offspring in the same generation.
2) Deb's feasibility rules [10] are added as to bias the search to the feasible region of the search space.
   a) Between 2 feasible vectors, the one with the highest fitness value wins.
   b) If one vector is feasible and the other one is infeasible, the feasible vector wins.
   c) If both vectors are infeasible, the one with the lowest sum of constraint violation is preferred ($\sum_{i=1}^{m} max(0, g_i(\vec{x}))$).
3) A selection ratio parameter $0 \leq S_r \leq 1$ is added to control the way vectors will be selected. Based on the S_r value the selection will be made based only in the value of the objective function $f(\vec{x})$, regardless of feasibility. Otherwise, the selection will be made based on the feasibility rules described before.

Based in the pseudocode in Figure 2, DDE adds two

```
Begin
  G=0
  Create a random initial population $\vec{x}_{i,G}$ $\forall i, i = 1, \dots, NP$
  Evaluate $f(\vec{x}_{i,G})$ $\forall i, i = 1, \dots, NP$
  For G=1 to MAX_GENERATIONS Do
    F=rand[0.3,0.9]
    For i=1 to NP Do
      For k=1 to $NO$ Do
        Select randomly $r_1 \neq r_2 \neq r_3 \neq i$
        $j_{rand}$ = randint$(1, D)$
        For j=1 to $n$ Do
          If $(rand_j[0,1) < CR$ or $j = j_{rand})$ Then
            child$_j = x_{r_3,j,G} + F(x_{r_1,j,G} - x_{r_2,j,G})$
          Else
            child$_j = x_{i,j,G}$
          End If
        End For
        If $k > 1$ Then
          If (child is better than $\vec{u}_{i,G+1}$
           based on the three selection criteria) Then
            $\vec{u}_{i,G+1}$=child
          End If
        Else
          $\vec{u}_{i,G+1}$=child
      End For
      If flip($S_r$) Then
        If $(f(\vec{u}_{i,G+1}) \leq f(\vec{x}_{i,G}))$ Then
          $\vec{x}_{i,G+1} = \vec{u}_{i,G+1}$
        Else
          $\vec{x}_{i,G+1} = \vec{x}_{i,G}$
        End If
      Else
        If ($\vec{u}_{i,G+1}$is better than $\vec{x}_{i,G}$
         based on the three selection criteria) Then
          $\vec{x}_{i,G+1} = \vec{u}_{i,G+1}$
        Else
          $\vec{x}_{i,G+1} = \vec{x}_{i,G}$
        End If
      End If
    End For
    $G = G + 1$
  End For
End
```

Fig. 2. DDE pseudocode. $randint(min, max)$ returns an integer value between min and max. rand$[0, 1)$ returns a real number between 0 and 1. Both functions adopt a uniform probability distribution. flip(W) returns 1 with probability W. NP, MAX_GEN, CR, F, NO and S_r are user-defined parameters. n is the dimensionality of the problem.

parameters (NO and S_r) to the original four parameters used in DE (NP, MAX_GEN, CR and F). Therefore, two parameter control mechanisms are proposed to keep the user from fine-tuning the values of four (out of six) parameters. Three parameters are self-adapted (CR, F and NO) and one of them uses a deterministic control (S_r). Furthermore, the behavior of these parameters are analyzed.

A. Self-adaptive parameter control

Motivated by the way Evolution Strategies work [23], three parameters are encoded in each solution: F, CR and NO as shown in Figure 3.

In this proposed approach, each solution has its own F, CR and NO values and they are subject to differential mutation and crossover. The process is explained in Figure 4, where the child vector in DDE will inherit the three parameter values from the target vector if the last decision variable was taken from it . On the other hand, the values for each parameter will be calculated by using the differential mutation operator i.e. they will be inherited from the mutation vector. The decision variables are handled as in traditional DE, however, the CR parameter value used in the process is the corresponding to the target vector.

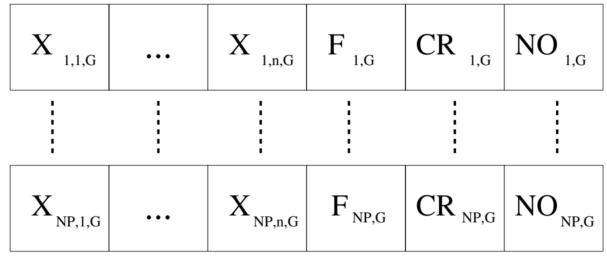

Fig. 3. Encoded solutions with three self-adapted parameters (F, CR and $N0$.

```
If (the last decision variable was inherited from the target vector) Then
  child$_j F$ = F$_{i,G}$
  child$_j CR$ = CR$_{i,G}$
  child$_j NO$ = NO$_{i,G}$
Else
  child$_j F$ = F$_{r_3,G}$ + F$_{i,G}$(F$_{r_1,G}$ − F$_{r_2,G}$)
  child$_j CR$ = CR$_{r_3,G}$ + F$_{i,G}$(CR$_{r_1,G}$ − CR$_{r_2,G}$)
  child$_j NO$ = NO$_{r_3,G}$ + F$_{i,G}$(NO$_{r_1,G}$ − NO$_{r_2,G}$)
End If
```

Fig. 4. Differential mutation applied to the self-adapted parameters. Note that the F value for the target vector $F_{i,G}$ is used in all cases.

B. Deterministic parameter control

Recalling the S_r parameter explanation, this parameter controls the comparisons made only by considering the objective function value, regardless of feasibility information. Therefore, it affects the bias in the search. Higher S_r values lead to keep infeasible solutions located in promising areas of the search space, whereas lower S_r values help to reach the feasible region by using Deb's rules [10].

Based on this behavior, the S_r parameter is controlled by a fixed schedule. A simple function is used to decrease the value for this parameter in such a way that initial higher values allow DDE to focus on searching promising regions of the search space, regardless of feasibility, with the aim to approach the feasible region from a more convenient area. Later in the process, the S_r values will be lower, assuming the feasible region has been reached and that it is more important to keep good feasible solutions. The interval within S_r values will be considered is the following: $[0.45, 0.65]$. At each generation, the S_r value will be decreased based on the following expression:

$$S_r^{(t+1)} = S_r^t - \Delta_{S_r} \tag{1}$$

where $S_r^{(t+1)}$ is the new value for this parameter, S_r^t is the current S_r value, Δ_{S_r} is the amount decreased from this value at each generation and calculated as follows:

$$\Delta_{S_r} = \left(\frac{S_r^0 - S_r^{G_{max}}}{G_{max}} \right) \tag{2}$$

where S_r^0 represents the initial value for S_r, and $S_r^{G_{max}}$ its last value in a given run.

The detailed pseudocode of Diversity Differential Evolution with the parameter control techniques, called Adaptive-DDE (A-DDE) is shown in Figure 5.

```
    Begin
      G=0
⇒     Create a random initial population X_{i,G} ∀i, i = 1, ..., NP
      Evaluate f(X_{i,G}) ∀i, i = 1, ..., NP
⇒     Select randomly SR ∈ [0.45, 0.65]
⇒     Select randomly SR_{G_max} ∈ (0.0, 0.45)
      For G=1 to G_max Do
        For i=1 to NP Do
⇒         For k=1 to NO_{i,G} Do
            Select randomly r_1 ≠ r_2 ≠ r_3 ≠ i
            j_rand = randint(1, D)
            For j=1 to D Do
⇒             If (rand_j[0, 1) < CR_{i,G} or j = j_rand) Then
                child_{j,G} = x_{r_3,j,G} + F_{i,G}(x_{r_1,j,G} − x_{r_2,j,G})
                ban=0
              Else
                child_{j,G} = x_{r_i,j,G}
                ban=1
              End If
            End For
⇒           If (ban==1) Then
              child_{F,G} = F_{i,G}
              child_{CR,G} = CR_{i,G}
              child_{NO,G} = NO_{i,G}
            Else
              child_{F,G} = F_{r_3,G} + F_{i,G}(F_{r_1,G} − F_{r_2,G})
              child_{CR,G} = CR_{r_3,G} + F_{i,G}(CR_{r_1,G} − CR_{r_2,G})
              child_{NO,G} = NO_{r_3,G} + F_{i,G}(NO_{r_1,G} − NO_{r_2,G})
            End If
            If k > 1 Then
              If (child is better than u_{i,G+1}
              (Based on three selection criteria))Then
                u_{i,G+1} = child
              End If
            Else
              u_{i,G+1} = child
            End If
          End For
          If flip(S_r)
            If (f(u_{i,G+1}) ≤ f(x_{i,G})) Then
              x_{i,G+1} = u_{i,G+1}
            Else
              x_{i,G+1} = x_{i,G}
            End If
          Else
            If (u_{i,G+1} ≤ x_{i,G}
            (Based on three selection criteria)) Then
              x_{i,G+1} = u_{i,G+1}
            Else
              x_{i,G+1} = x_{i,G}
            End If
          End If
        End For
        G = G + 1
⇒       S_r = S_r − Δ_{S_r}
      End For
    End
```

Fig. 5. A-DDE pseudocode. Arrows indicate steps where the parameter control mechanisms are involved.

IV. Experiments and Results

The experiments are designed: (1) to evaluate if the proposed parameter control does not affect the performance of DDE, (2) to determine if the proposed approach is not similar to just generating random values within recommended intervals (3) to analyze the values taken by the parameters and (4) to compare the performance of A-DDE against state-of-the-art approaches.

13 test problems taken from the specialized literature [21] are used in all the experiments. A summary of their features is shown in Table I. 30 independent runs were performed in all the experiments.

TABLE I

Main features of test problems. ρ is the estimated size of the feasible region with respect to the whole search space, LI and NI are the number of linear and nonlinear inequality constraints respectively and LE and NE are the number of linear and nonlinear equality constraints. Finally A indicates the number of active constraints.

P	N	Function	ρ	LI	NI	LE	NE	A
g01	13	quadratic	0.0111%	9	0	0	0	6
g02	20	nonlinear	99.9971%	0	2	0	0	1
g03	10	polynomial	0.0000%	0	0	0	1	1
g04	5	quadratic	52.1230%	0	6	0	0	2
g05	4	cubic	0.0000%	2	0	0	3	3
g06	2	cubic	0.0066%	0	2	0	0	2
g07	10	quadratic	0.0003%	3	5	0	0	6
g08	2	nonlinear	0.8560%	0	2	0	0	0
g09	7	polynomial	0.5121%	0	4	0	0	2
g10	8	linear	0.0010%	3	3	0	0	6
g11	2	quadratic	0.0000%	0	0	0	1	1
g12	3	quadratic	4.7713%	0	1	0	0	0
g13	5	nonlinear	0.0000%	0	0	0	3	3

In the first experiment the original DDE and A-DDE are compared. The parameters used for each algorithm were the following:

1) DDE
   - $NP = 60$ and $GMAX = 600$
   - $S_r = 0.45$
   - $NO = 5$, $CR = 0.9$ and $F \in [0.3, 0.9]$ generated at random.
2) A-DDE.
   - $NP = 60$ and $GMAX = 600$
   - $S_r \in [0.45, 0.65]$, $S_r^{G_{max}} \in (0.0, 0.45)$, randomly generated on each independent run and this value is controlled by the deterministic control.
   - $NO \in [3, 7]$, $CR \in [0.9, 1.0]$ and $F \in [0.3, 0.9]$ initially generated at random and then handled with the self-adaptive control.

The number of evaluations performed by each approach is $180,000$ in order to promote a fair comparison. The statistical results are summarized in Table II.

Based on the results in Table II, A-DDE maintains the performance obtained by the original DDE. It is worth noticing that in problem g10 the robustness of A-DDE (i.e. the mean and worst values) is slightly better. On the other hand, only the best value in problem $g02$ is slightly affected as well as the robustness (mean and worst values) in problems $g02$ and $g13$.

The second experiment compares A-DDE with a DDE version (called R-DDE) where random values for the four parameters analyzed in this paper are generated. The aim of this experiment is to be sure that the adaptive and deterministic control mechanisms proposed are not equivalent to the simple generation of random values within the suggested intervals. The parameters used in R-DDE were the following:

- R-DDE
  - $NP = 60$ and $GMAX = 600$

TABLE II

COMPARISON OF RESULTS WITH THE ORIGINAL DDE VERSION (STATIC PARAMETER VALUES) AND THE PROPOSED A-DDE (ADAPTIVE AND SELF-ADAPTIVE PARAMETER VALUES). VALUES IN **boldface** REPRESENT THAT THE GLOBAL OPTIMUM OR BEST KNOW SOLUTION WAS REACHED. A VALUE IN *italic* MEANS THAT THE RESULT IS BETTER THAN THE OBTAINED WITH THE COMPARED APPROACH.

Test problem	Best known solution	Best		Mean		Worst	
		A-DDE	DDE	A-DDE	DDE	A-DDE	DDE
g01	-15.000	**-15.000**	**-15.000**	**-15.000**	**-15.000**	**-15.000**	**-15.000**
g02	-0.803619	-0.803605	*-0.803618*	-0.771090	*-0.789132*	-0.609853	*-0.747876*
g03	-1.000	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**
g04	-30665.539	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**
g05	5126.497	**5126.497**	**5126.497**	**5126.497**	**5126.497**	**5126.497**	**5126.497**
g06	-6961.814	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**
g07	24.306	**24.306**	**24.306**	**24.306**	**24.306**	**24.306**	**24.306**
g08	-0.095825	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**
g09	680.63	**680.63**	**680.63**	**680.63**	**680.63**	**680.63**	**680.63**
g10	7049.248	**7049.248**	**7049.248**	**7049.248**	7049.262	**7049.248**	7049.503
g11	0.75	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**
g12	-1.000	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**
g13	0.053942	**0.053942**	**0.053942**	0.079627	**0.053942**	0.438803	*0.053961*

- $S_r \in [0.45, 0.65]$ generated at random instead of using the deterministic parameter control.
- $NO \in [3, 7]$, $CR \in [0.9, 1.0]$ and $F \in [0.3, 0.9]$ also generated at random instead of using the self-adaptive parameter control.

The number of evaluations was the same used in the previous experiment $(180,000)$. The comparison of results between A-DDE and R-DDE is shown in Table III

The statistics in Table III show that the quality of results (the best result found so far) is not affected in R-DDE. However, the robustness is clearly deteriorated in several test functions ($g02$, $g04$, $g06$, $g07$, $g09$, $g10$ and $g13$). This behavior suggests that sometimes adequate values are chosen (randomly) for the parameters in R-DDE. Nonetheless, poor choices are taken in other times. In contrast, A-DDE finds in each single run the adequate values to keep competitive results in most independent runs.

The third experiment aims to analyze the behavior of the self-adaptive parameter control on A-DDE. The convergence graphs for the three compared approaches and also two graphs showing the average value for each parameter in the population at each generation in A-DDE are presented for representative problems. In all cases, the graphs are based on the run located at the median value, out of the 30 independent runs. Three test problems were selected based on the behavior observed: $g04$ where A-DDE converges in a similar way with respect to DDE and R-DDE (see Figure 6), $g10$ where A-DDE converges faster to a better solution with respect to the two compared approaches (see Figure 7) and $g02$, where A-DDE got trapped in a local optima solution (see Figure 8). Regarding problem $g04$ (Figure 6), the CR parameter values slightly oscillate until convergence is reached. Almost in the same way the F parameter values behave, but the oscillation is more evident. The $N0$ parameter conserves the value of 4 almost all the time before convergence, but it scales its value to 5 more frequently than going down to 3. In problem $g10$ (Figure 7), the CR, F and $N0$ values never converge. Instead, they keep oscillating during all the process. $N0$, as in problem $g04$, tends to take the value of 5 more frequently with respect to the value of 4. Finally, in problem $g02$ (Figure 8), the oscillating behavior is not observed. In contrast, the values are maintained during several generations and some remarked differences in the values are observed at the end of the process, mostly for parameter F. The $N0$ parameter moved more times to the value of 3 (the opposite with respect to the two previous test problems). The overall results in this experiment show that the CR parameter requires small variations to its high value (near 0.9) during the process. This means that the trial vector are generated most of the time with values from the mutation vector instead of the target vector. For the F parameter, the variation is more remarked, which means that the scaled search directions must be more diverse and, if the scale remains fixed for some generations, the algorithm can be trapped in local optima. Finally, the number of offspring generated per each target vector requires some dispersed increments from time to time as to also increase the chances of getting a better trial vector, but a value of 4 seems to be adequate.

As a final experiment, we compare the results obtained by A-DDE with respect to three state-of-the-art approaches: The adaptive trade-off model by Wang et al. [24], the adaptive penalty function by Tessema & Yen [21] and a mathematical programming approach combined with a mutation operator by Takahama & Sakai [25]. Based on the comparison in table IV it is clear that A-DDE keeps the competitive performance with respect to those provided by other proposed techniques found in the specialized literature.

It is important to remark that A-DDE requires the direct definition of only two parameters (NP and G_{max}), and its computational cost is quite low: $180,000$ evaluations compared to $500,000$ and $240,000$ evaluations required by Tessema & Yen [21] and Wang et al. [24], respectively. The approach proposed by Takahama and Sakai [25] separates the evaluation of objective function and constraints, but it requires between $290,000$ and $330,000$ evaluations (on average) for the constraints in 12 of 13 problems.

TABLE III

COMPARISON OF RESULTS WITH R-DDE VERSION (RANDOM PARAMETER VALUES) AND THE PROPOSED A-DDE. VALUES IN **boldface** REPRESENT THAT THE GLOBAL OPTIMUM OR BEST KNOW SOLUTION WAS REACHED. A VALUE IN *italic* MEANS THAT THE RESULT IS BETTER THAN THE OBTAINED WITH THE COMPARED APPROACH.

Test problem	Best known solution	**Best**		**Mean**		**Worst**	
		A-DDE	R-DDE	A-DDE	R-DDE	A-DDE	R-DDE
g01	-15.000	**-15.000**	**-15.000**	**-15.000**	-14.937	**-15.000**	-13.917
g02	-0.803619	-0.803605	**-0.803610**	*-0.771090*	-0.706674	*-0.609853*	-0.483550
g03	-1.000	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**
g04	-30665.539	**-30665.539**	**-30665.539**	**-30665.539**	-30660.237	**-30665.539**	-30591.889
g05	5126.497	**5126.497**	**5126.497**	**5126.497**	**5126.497**	**5126.497**	**5126.497**
g06	-6961.814	**-6961.814**	**-6961.814**	**-6961.814**	-6959.015	**-6961.814**	-6877.840
g07	24.306	**24.306**	**24.306**	**24.306**	24.945	**24.306**	38.903
g08	-0.095825	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**
g09	680.63	**680.63**	**680.63**	**680.63**	**680.63**	**680.63**	**680.63**
g10	7049.248	**7049.248**	**7049.248**	**7049.248**	7073.779	**7049.248**	7308.826
g11	0.75	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**
g12	-1.000	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**
g13	0.053942	**0.053942**	**0.053942**	*0.079627*	0.131458	*0.438803*	0.438803

TABLE IV

STATISTICAL RESULTS OBTAINED BY A-DDE WITH RESPECT TO THOSE PROVIDED BY STATE-OF-THE-ART APPROACHES ON 13 BENCHMARK PROBLEMS. BEST RESULTS ARE REMARKED IN **boldface**.

Problem/BKS	Statistic	Wang et al. [24]	Tessema & Yen [21]	Takahama & Sakai [25]	A-DDE
g01	Best	-15.000	-15.000	-15.000	-15.000
-15.000	Median	-15.000	-14.966	-15.000	-15.000
	Worst	-15.000	-13.097	-15.000	-15.000
	St. Dev.	**1.60E-14**	7.00E-01	6.40E-06	7.00E-06
g02	Best	0.803338	0.803202	**0.803619**	0.803605
-0.803619	Median	**0.792420**	0.789398	0.785163	0.777368
	Worst	**0.756986**	0.745712	0.754259	0.609853
	St. Dev.	**1.30E-02**	1.33E-01	1.30E-02	3.66E-02
g03	Best	1.000	1.000	1.000	1.000
-1.0005	Median	1.000	0.971	1.000	1.000
	Worst	1.000	0.887	1.000	1.000
	St. Dev.	5.90E-05	3.01E-01	**8.50E-14**	9.30E-12
g04	Best	-30665.539	-30665.401	-30665.539	-30665.539
-30665.539	Median	-30665.539	-30663.921	-30665.539	-30665.539
	Worst	-30665.539	-30656.471	-30665.539	-30665.539
	St. Dev.	7.40E-12	2.04E+00	4.20E-11	**3.20E-13**
g05	Best	5126.498	5126.907	5126.497	5126.497
5126.497	Median	5126.776	5208.897	5126.497	5126.497
	Worst	5135.256	5564.642	5126.497	5126.497
	St. Dev.	1.80E+00	2.47E+02	3.50E-11	**2.10E-11**
g06	Best	-6961.814	-6961.046	-6961.814	-6961.814
-6961.814	Median	-6961.814	-6953.823	-6961.814	-6961.814
	Worst	-6961.814	-6943.304	-6961.814	-6961.814
	St. Dev.	4.60E-12	5.88E+00	1.30E-10	**2.11E-12**
g07	Best	24.306	24.838	24.306	24.306
24.306	Median	24.313	25.415	24.306	24.306
	Worst	24.359	33.095	24.307	**24.306**
	St. Dev.	1.10E-02	2.17E+00	**1.30E-04**	**4.20E-05**
g08	Best	0.095825	0.095825	0.095825	0.095825
-0.095825	Median	0.095825	0.095825	0.095825	0.095825
	Worst	0.095825	0.092697	0.095825	0.095825
	St. Dev.	**2.80E-17**	1.06E-03	3.80E-13	9.10E-10
g09	Best	680.63	680.77	680.63	680.63
680.63	Median	680.63	681.24	680.63	680.63
	Worst	680.67	682.08	680.63	680.63
	St. Dev.	1.00E-02	3.22E-01	2.90E-10	**1.15E-10**
g10	Best	7052.253	7069.981	7049.248	7049.248
7049.248	Median	7215.357	7201.017	7049.248	7049.248
	Worst	7560.224	7489.406	7049.248	7049.248
	St. Dev.	1.20E+02	1.38E+02	**4.70E-06**	3.23E-4
g11	Best	0.75	0.75	0.75	0.75
0.75	Median	0.75	0.75	0.75	0.75
	Worst	0.75	0.76	0.75	0.75
	St. Dev.	3.40E-04	2.00E-03	**4.90E-16**	5.35E-15
g12	Best	1.000	1.000	1.000	1.000
-1.000	Median	1.000	1.000	1.000	1.000
	Worst	0.994	1.000	1.000	1.000
	St. Dev.	1.00E-03	1.41E-04	3.90E-10	**4.10E-11**
g13	Best	0.053950	**0.053941**	0.053942	0.053942
0.053842	Median	0.053952	0.054713	0.053942	0.053942
	Worst	**0.053999**	0.885276	0.438803	0.438803
	St. Dev.	**1.30E-05**	2.75E-01	6.90E-02	9.60E-02

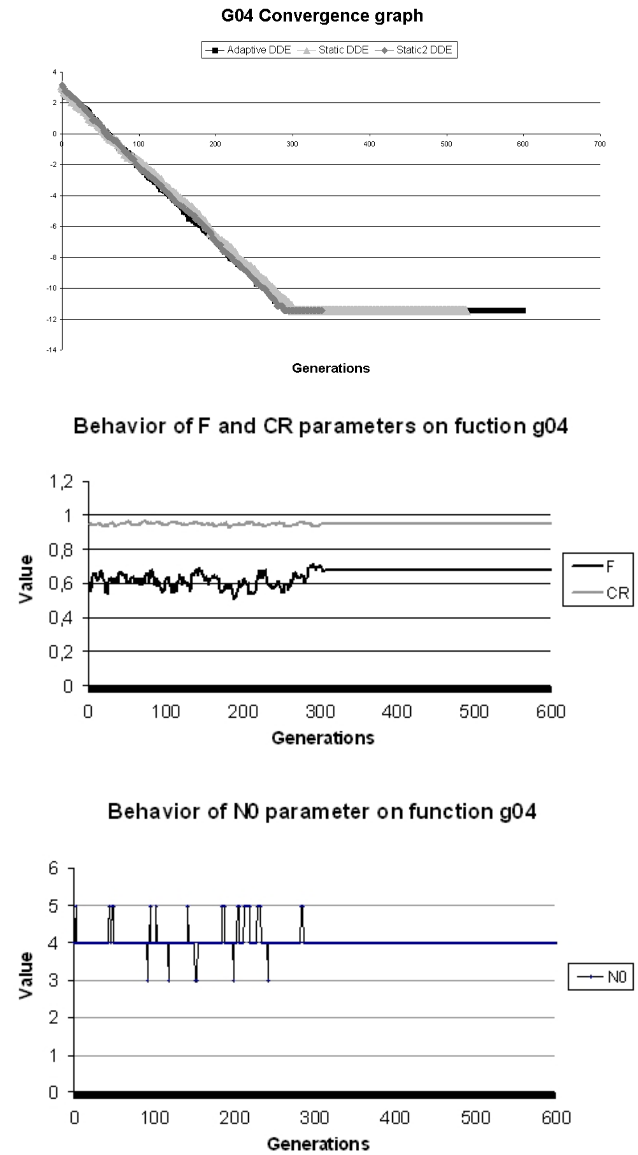

Fig. 6. Convergence and parameter graphs for problems g04.

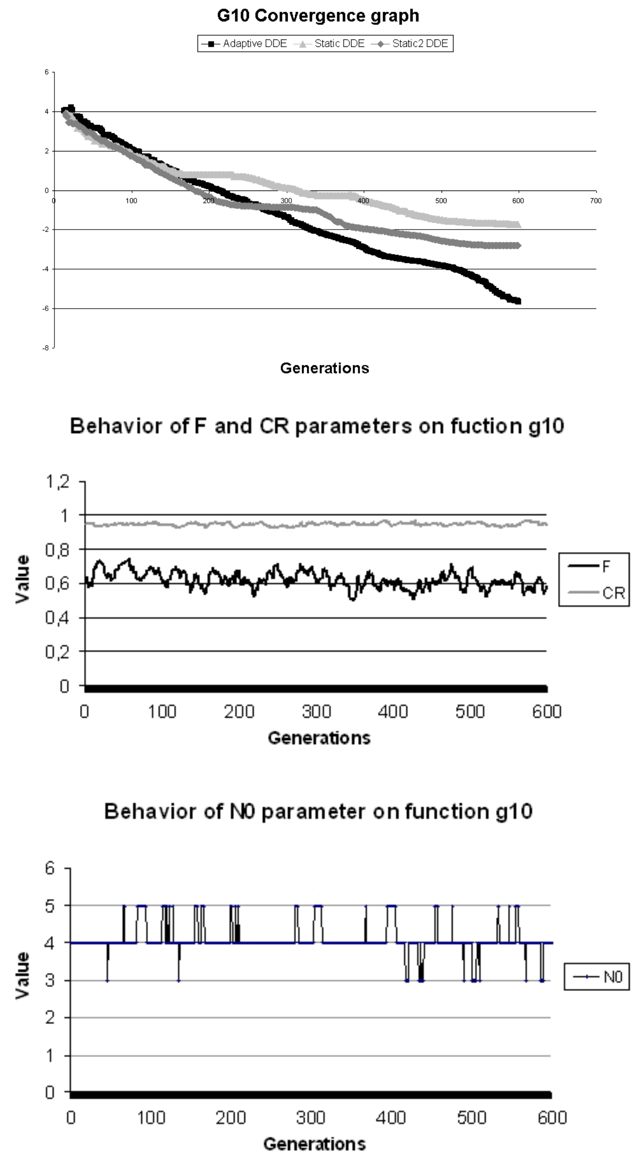

Fig. 7. Convergence and parameter graphs for problems g10.

V. Conclusions and Future Work

A proposal to incorporate parameter control mechanisms into a DE-based approach (called A-DDE) for constrained optimization was presented. Three parameters: F, CR and $N0$ (the number of trial vectors per each target vector) were encoded within each solution and subject to a self-adaptive parameter control. Another parameter, S_r, which handles the diversity in the population, was controlled by a deterministic approach. Four experiments were performed (1) to verify that the proposed parameter control does not affect the performance of DDE, (2) to analyze that the proposed parameter control is not similar to the generation of random values for each one of the parameters, (3) to study the behavior of each parameter and (4) to compare the obtained results with some approaches found in the specialized literature. The findings of this research provided information about the values each parameter requires at different stages of the optimization process and give to interested practitioners and researchers a competitive approach where just two parameters may be fine-tuned. Part of our future work is to use performance measures as to know the on-line behavior of A-DDE i.e. how fast it reaches the feasible region and how capable is to sample it. Finally, we will test A-DDE in real-world constrained optimization problems.

Acknowledgment

The first author acknowledges support from CONACyT through project No. 79809. The second authors acknowledges support from CONACyT through a scholarship.

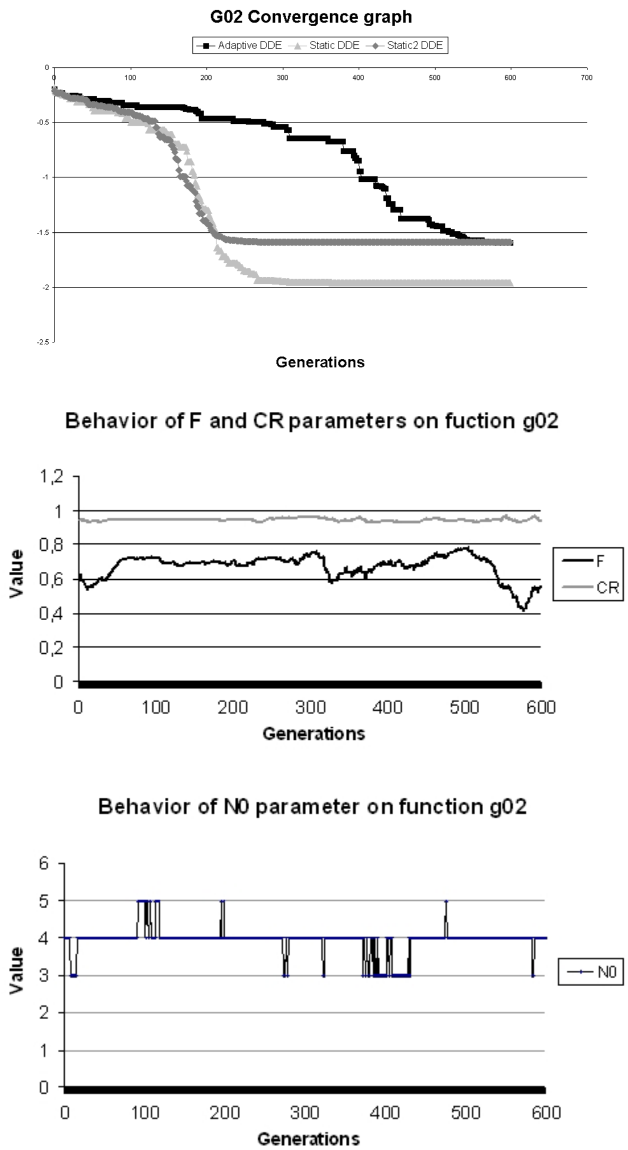

Fig. 8. Convergence and parameter graphs for problems g02.

References

[1] Z. Michalewicz and D. B. Fogel, *How to Solve It: Modern Heuristics*, 2nd ed. Germany: Springer, 2004.

[2] A. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, ser. Natural Computing Series. Springer Verlag, 2003.

[3] E. Mezura-Montes, Ed., *Constraint-Handling in Evolutionary Optimization*, ser. Studies in Computational Intelligence. Berlin Heideberg: Springer-Verlag, 2009, vol. 198.

[4] C. A. C. Coello, "Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, January 2002.

[5] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, ser. Natural Computing Series. Springer-Verlag, 2005.

[6] U. Chakraborty, Ed., *Advances in Differential Evolution*, ser. Studies in Computational Intelligence. Berlin Heideberg: Springer-Verlag, 2008, vol. 143.

[7] G. Eiben and M. Schut, "New Ways to Calibrate Evolutionary Algorithms," in *Advances in Metaheuristics for Hard Optimization*, ser. Natural Computing, P. Siarry and Z. Michalewicz, Eds. Heidelberg, Germany: Springer, 2008, pp. 153–177.

[8] J. Lampinen, "A Constraint Handling Approach for the Diifferential Evolution Algorithm," in *Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)*, vol. 2. Piscataway, New Jersey: IEEE Service Center, May 2002, pp. 1468–1473.

[9] E. Mezura-Montes, C. A. Coello Coello, and E. I. Tun-Morales, "Simple Feasibility Rules and Differential Evolution for Constrained Optimization," in *Proceedings of the 3rd Mexican International Conference on Artificial Intelligence (MICAI'2004)*, R. Monroy, G. Arroyo-Figueroa, L. E. Sucar, and H. Sossa, Eds. Heidelberg, Germany: Springer Verlag, April 2004, pp. 707–716, lecture Notes in Artificial Intelligence No. 2972.

[10] K. Deb, "An Efficient Constraint Handling Method for Genetic Algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2/4, pp. 311–338, 2000.

[11] S. Kukkonen and J. Lampinen, "Constrained Real-Parameter Optimization with Generalized Differential Evolution," in *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*. Vancouver, BC, Canada: IEEE, July 2006, pp. 911–918.

[12] K. Zielinski and R. Laur, "Constrained Single-Objective Optimization Using Differential Evolution," in *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*. Vancouver, BC, Canada: IEEE, July 2006, pp. 927–934.

[13] T. Takahama and S. Sakai, "Constrained Optimization by the ϵ Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites," in *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*. Vancouver, BC, Canada: IEEE, July 2006, pp. 308–315.

[14] ——, "Constrained optimization by ϵ-constrained differential evolution with dynamic ϵ-level control," in *Advances in Differential Evolution*, ser. Studies in Computational Intelligence, U. Chakraborty, Ed. Berlin Heidelberg: Springer, 2008, pp. 139–154.

[15] A. E. Muñoz-Zavala, A. Hernández-Aguirre, E. R. Villa-Diharce, and S. Botello-Rionda, "PESO+ for Constrained Optimization," in *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*. Vancouver, BC, Canada: IEEE, July 2006, pp. 935–942.

[16] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello, "Modified Differential Evolution for Constrained Optimization," in *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*. Vancouver, BC, Canada: IEEE, July 2006, pp. 332–339.

[17] M. F. Tasgetiren and P. N. Suganthan, "A Multi-Populated Differential Evolution Algorithm for Solving Constrained Optimization Problem," in *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*. Vancouver, BC, Canada: IEEE, July 2006, pp. 340–354.

[18] J. Brest, V. Žumer, and M. S. Maučecc, "Control Parameters in Self-Adaptive Differential Evolution," in *Bioinspired Optimization Methods and Their Applications*, B. Filipič and J. Šilc, Eds. Ljubljana, Slovenia: Jožef Stefan Institute, October 2006, pp. 35–44.

[19] V. L. Huang, A. K. Qin, and P. N. Suganthan, "Self-adaptative Differential Evolution Algorithm for Constrained Real-Parameter Optimization," in *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*. Vancouver, BC, Canada: IEEE, July 2006, pp. 324–331.

[20] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Comput.*, vol. 9, no. 6, pp. 448–462, 2005.

[21] B. Tessema and G. G. Yen, "A Self Adaptative Penalty Function Based Algorithm for Constrained Optimization," in *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*. Vancouver, BC, Canada: IEEE, July 2006, pp. 950–957.

[22] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello, "Promising Infeasibility and Multiple Offspring Incorporated to Differential Evolution for Constrained Optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2005)*, vol. 1. New York: ACM Press, June 2005, pp. 225–232.

[23] H.-P. Schwefel, Ed., *Evolution and Optimization Seeking*. New York: Wiley, 1995.

[24] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An Adaptive Tradeoff Model for Constrained Evolutionary Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 80–92, February 2008.

[25] T. Takahama and S. Sakai, "Constrained Optimization by Applying the α Constrained Method to the Nonlinear Simplex Method with Mutations," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 437–451, October 2005.